

# Hinweise zum Upgrade auf LabVIEW™ 2017

In diesem Dokument wird beschrieben, wie Sie LabVIEW für Windows, OS X und Linux auf LabVIEW 2017 aktualisieren. Bevor Sie auf die neue LabVIEW-Version umsteigen, informieren Sie sich in diesem Dokument zu den folgenden Themen:

- die empfohlene Vorgehensweise für das Upgrade von LabVIEW
- Hinweise zu möglichen Kompatibilitätsproblemen beim Laden von VIs, die mit älteren Versionen von LabVIEW erstellt wurden
- Änderungen und Neuerungen in LabVIEW 2017

## Inhalt

---

Upgrade auf LabVIEW 2017.....	1
1. Sichern der VIs und der Computerkonfiguration.....	2
2. Prüfen der aktuellen Funktionsweise der VIs.....	3
3. Installieren von LabVIEW sowie von Zusatzpaketen und Gerätetreibern.....	4
4. Umwandeln der VIs und Beseitigen von Funktionsunterschieden.....	4
Fehlersuche beim Upgrade.....	6
Hinweise zur Kompatibilität beim Upgrade.....	6
Upgrade von LabVIEW 2012 oder älteren Versionen.....	6
Upgrade von LabVIEW 2013.....	6
Upgrade von LabVIEW 2014.....	8
Upgrade von LabVIEW 2015.....	8
Upgrade von LabVIEW 2016.....	8
Änderungen und Neuerungen in LabVIEW 2017.....	9
Reduzierte Lade- und Kompilierzeit für VIs.....	9
Beibehalten von Verbindungen beim Verschieben von Objekten.....	9
Datentypflexible VIs.....	9
Neue und geänderte VIs und Funktionen.....	10
Neue und geänderte Klassen, Eigenschaften, Methoden und Ereignisse.....	10
Verbesserungen am Application Builder.....	11
Änderungen und Neuerungen in vorherigen LabVIEW-Versionen.....	12

## Upgrade auf LabVIEW 2017

---

Zum Umstellen kleinerer Anwendungen auf eine neue LabVIEW-Version genügt es in der Regel, die VIs in der neuen LabVIEW-Version zu laden. Um durch die Umstellung verursachte Probleme so effizient wie möglich zu erkennen und zu beheben, wird jedoch eine systematischere Vorgehensweise empfohlen.



**Tipp** Große LabVIEW-Anwendungen zur Steuerung und Überwachung von kritischen Operationen sowie Anwendungen, deren Ausfall kostspielig ist, profitieren insbesondere von der systematischen Umstellung. Das Gleiche gilt für Anwendungen, die mit mehreren Modulen und Toolkits arbeiten oder die in einer nicht unterstützten LabVIEW-Version gespeichert sind. Informationen zu LabVIEW-Versionen, zu denen Standard-Support erhältlich ist, finden Sie auf der NI-Webseite unter [ni.com/info](http://ni.com/info) nach Eingabe des Infocodes `lifecycle`.

## Überblick über den empfohlenen Upgrade-Prozess

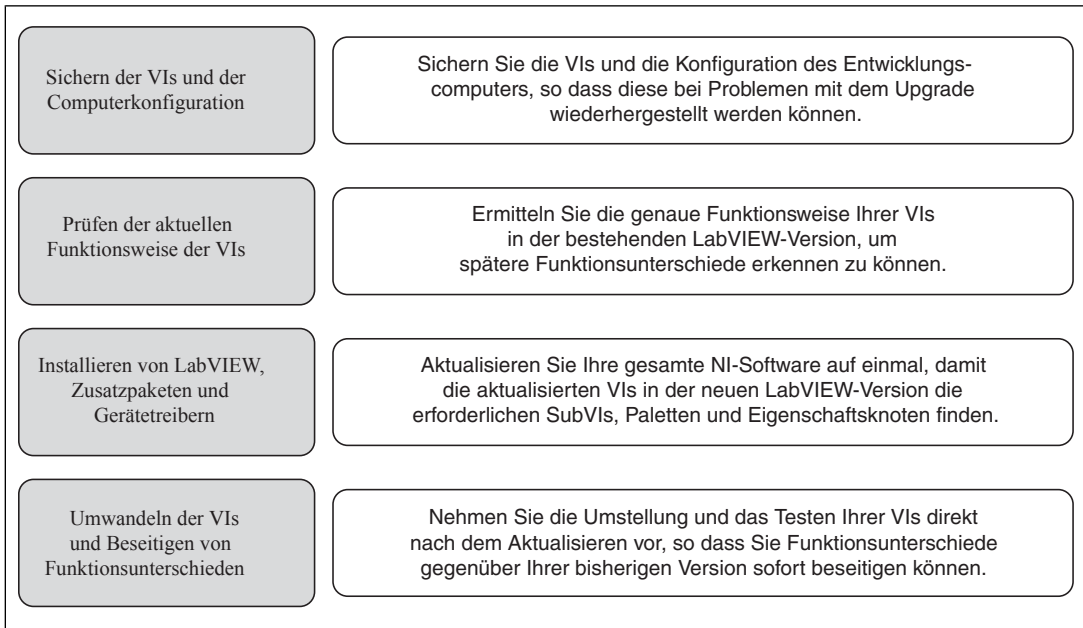


Figure 1.



**Hinweis** Zum Umstellen Ihrer VIs von einer LabVIEW-Version bis einschließlich 5.1 müssen Sie die VIs zuerst auf eine Zwischenversion von LabVIEW aktualisieren. Weitere Informationen zum Upgrade einer älteren LabVIEW-Version erhalten Sie auf der Website [ni.com/info](http://ni.com/info) nach Eingabe des Infocodes `upgradeOld`.

### 1. Sichern der VIs und der Computerkonfiguration

Durch Sichern einer Kopie der VIs und – sofern möglich – der Konfiguration des Computers vor dem Upgrade auf LabVIEW 2017 wird gewährleistet, dass Sie den Upgrade-Prozess bei Bedarf erneut beginnen können.

#### a. Sichern von VIs

Wenn Sie Ihre VIs vor dem LabVIEW-Upgrade sichern, können Sie die alte Version problemlos wiederherstellen. Ohne die Sicherungskopie können Sie die aktualisierten VIs nicht mehr in älteren LabVIEW-Versionen öffnen. Die VIs müssen dann erst in der neuen Version für eine ältere Version gespeichert werden.

Die VIs können auf folgende Art und Weise gesichert werden:

- **Speichern der VIs in einem Versionsverwaltungssystem**—Ermöglicht das Wiederherstellen der vorherigen VI-Version, wenn sich durch die Umstellung verursachte Funktionsunterschiede nicht beheben lassen. Weitere Informationen zur Verwendung von Versionsverwaltungssystemen mit LabVIEW finden Sie auf der Registerkarte **Inhalt** der *LabVIEW-Hilfe* unter **Grundlagen»Arbeiten mit Projekten und Zielen»Allgemeines»Verwendung der Versionsverwaltung in LabVIEW**.

- **Erstellen einer Kopie der VIs**— Gehen Sie zum Erstellen einer Sicherungskopie je nach der ursprünglichen Speicher­methode der VIs wie folgt vor:
  - Als Projekt gespeichert—Öffnen Sie das Projekt und speichern Sie über **Datei»Speichern** unter eine Kopie der \*.lvproj-Datei und des Projektinhalts. Wählen Sie **Alle Abhängigkeiten einschließen**, um auch Komponenten einzuschließen, von denen das Projekt abhängt.
  - Als LLB oder in einem Verzeichnis gespeichert—Erstellen Sie im Datei-Explorer Ihres Betriebssystems eine Kopie der LLB oder des Verzeichnisses und speichern Sie diese an einer anderen Stelle. Zum Vermeiden von Namenskonflikten sollte die Kopie nicht auf derselben Festplatte gespeichert werden.

## b. Sichern der Computerkonfiguration

Beim Upgrade von LabVIEW werden versionsübergreifend verwendete gemeinsame Dateien verändert. Das kann in Ausnahmefällen dazu führen, dass Ihre VIs in älteren Versionen nicht mehr erwartungsgemäß funktionieren. Die vorherige Version der gemeinsam genutzten Dateien kann nach dem Installieren der neuen LabVIEW-Version nicht ohne Weiteres wiederhergestellt werden. Daher sollten Sie die Konfiguration der NI-Software auf Ihrem Entwicklungscomputer mit einer der nachfolgend aufgeführten Methoden sichern. Das gilt insbesondere, wenn Ihre bisher verwendete LabVIEW-Version nicht mehr unterstützt wird oder ein Ausfall Ihrer Anwendungen kostspielig wäre.

- **Erstellen eines Abbilds der Computerkonfiguration**—Erstellen Sie vor dem Umstellen auf die neue Version mit Hilfe von *Disk-Imaging-Software* ein Festplattenabbild des Computers, das die Dateien, die installierte Software sowie sämtliche Benutzereinstellungen enthält. Mit dem Festplattenabbild können Sie den Computer nach dem Upgrade ggf. wieder in den Originalzustand zurückversetzen.
- **Testen Sie den Upgrade-Prozess auf einem Testcomputer**—Obwohl das LabVIEW-Upgrade auf einem Testcomputer gegenüber dem Erstellen eines Festplattenabbilds zeitaufwändiger ist, wird diese Vorgehensweise dennoch dringend für alle ausfallkritischen Systeme in Produktionsprozessen empfohlen. Nach dem Beheben eventueller Upgrade-Probleme auf dem Testcomputer können Sie den Produktionscomputer entweder durch den Testcomputer ersetzen oder den Upgrade-Prozess auf dem Produktionscomputer wiederholen.



**Tipp** Damit VIs auf dem Testcomputer genauso funktionieren wie auf dem später genutzten Computer, sollte die Konfiguration der Computer (u. a. hinsichtlich CPU, RAM, Betriebssystem und Softwareversionen) möglichst ähnlich sein.

## 2. Prüfen der aktuellen Funktionsweise der VIs

Die Weiterentwicklung von LabVIEW kann gelegentlich dazu führen, dass gegenüber der Vorgängerversion von LabVIEW Funktionsunterschiede in VIs auftreten. Testen Sie die VIs daher in der bisherigen und der neuen LabVIEW-Version. Vergewissern Sie sich, dass Ihnen aktuelle Ergebnisse für die folgenden Tests vorliegen:

- Führen Sie eine Massenkompilierung der VIs in der Vorgängerversion von LabVIEW durch und prüfen Sie das dabei erstellte Massenkompilierungsprotokoll auf fehlerhafte VIs. Eine Massenkompilierung von VIs vor dem Upgrade ist besonders nützlich, wenn mehrere Personen an der VI-Entwicklung beteiligt sind oder wenn Sie vermuten, dass VIs lange nicht mehr kompiliert wurden. Damit ein Massenkompilierungsprotokoll erstellt wird, aktivieren Sie die Option **Ergebnisse aufzeichnen** im Dialogfeld **Massenkompilierung**. Weitere Informationen dazu finden Sie in der *LabVIEW-Hilfe* unter **Grundlagen»Erstellen von VIs und SubVIs»Anleitung»Speichern von VIs»Massenkompilierung von VIs**.
- Prüfen Sie mit Modultests, ob einzelne VIs ihre vorgesehenen Funktionen erfüllen.

- Prüfen Sie mit Integrationstests, ob SubVIs in einem Projekt oder einer Gruppe erwartungsgemäß zusammenarbeiten.
- Prüfen Sie mit Einsatztests, ob die VIs nach dem Übertragen auf ein Zielsystem (z. B. einen Desktop-Computer oder FPGA-Chip) erwartungsgemäß funktionieren.
- Prüfen Sie mit Leistungstests die CPU-Auslastung, den Speicherbedarf und die Ausführungsgeschwindigkeit des Programmcodes. Im Fenster **Profil - Leistung und Speicher** finden Sie Schätzwerte für die durchschnittliche Ausführungsgeschwindigkeit der VIs.
- Prüfen Sie mit Stresstests, ob die VIs unerwartete Daten korrekt verarbeiten.

Weitere Informationen zum Testen der VIs finden Sie in der *LabVIEW-Hilfe* unter **Grundlagen»Entwicklungsrichtlinien für Anwendungen»Allgemeines»Entwicklung großer Anwendungen»Phasen von Entwicklungsmodellen»Testen von Anwendungen**.



**Hinweis** Wenn sich VIs nach der Massenkompilierung oder nach Tests geändert haben, sollten Sie zunächst die neuen Versionen der VIs sichern.

### 3. Installieren von LabVIEW sowie von Zusatzpaketen und Gerätetreibern

#### a. Installieren von LabVIEW einschließlich Modulen, Toolkits und Treibern

Beim Umsteigen auf eine neue LabVIEW-Version müssen Sie neben dem LabVIEW Development System auch Module, Toolkits und Treiber für die neue LabVIEW-Version installieren.

#### b. Kopieren von user.lib-Dateien

Damit alle benutzerdefinierten Elemente und VIs aus der bisherigen LabVIEW-Version auch für die VIs in LabVIEW 2017 verfügbar sind, müssen Sie das Verzeichnis `labview\user.lib` Ihrer bisherigen Version in das Verzeichnis `labview\user.lib` von LabVIEW 2017 kopieren.

### 4. Umwandeln der VIs und Beseitigen von Funktionsunterschieden

VIs werden durch eine Massenkompilierung in LabVIEW 2017 auf diese LabVIEW-Version umgestellt. Das dabei erzeugte Fehlerprotokoll enthält Angaben zu fehlerhaften VIs. Werten Sie diese Angaben anhand des Abschnitts *Hinweise zur Kompatibilität beim Upgrade* in diesem Dokument aus, um durch das Upgrade verursachte Funktionsunterschiede zu erkennen.

#### a. Massenkompilierung von VIs in der neuen Version von LabVIEW

Bei der Massenkompilierung werden die VIs gleichzeitig konvertiert und in LabVIEW 2017 gespeichert. Danach können die VIs jedoch nicht mehr in einer Vorgängerversion von LabVIEW geöffnet werden. Zum Öffnen der VIs in einer Vorgängerversion muss für jedes VI oder Projekt die Option **Datei»Für vorige Version speichern** gewählt werden. Daher sollten VIs nur dann massenkompiliert werden, wenn sie auf die neue Version umgestellt werden sollen. Zum Erkennen von aktualisierungsbezogenen Problemen führen Sie eine Massenkompilierung mit Fehlerprotokoll durch, indem Sie im Dialogfeld **Massenkompilierung** die Option **Ergebnisse aufzeichnen** aktivieren.



**Hinweis** Bei der Massenkompilierung von VIs, die FPGA- oder Real-Time-Ressourcen enthalten, werden die VIs im Dialogfeld **Massenkompilierung** unter Umständen als "nicht ausführbar" angezeigt. Um die VIs auf Fehler zu prüfen, müssen Sie sie auf dem FPGA- oder Real Time-Zielsystem in einem LabVIEW-Projekt mit den erforderlichen FPGA- oder Real-Time-Ressourcen öffnen.

Informationen zur Massenkompilierung finden Sie in den folgenden Abschnitten der *LabVIEW-Hilfe*:

- **Grundlagen»Erstellen von VIs und SubVIs»Anleitung»Speichern von VIs»Massenkompilierung von VIs**

- **Grundlagen»Erstellen von VIs und SubVIs»Anleitung»Speichern von VIs»Gängige Fehlermeldungen bei der Massenkompilierung**

## b. Korrigieren von Fehlern in VIs

Änderungen zwischen Ihrer bisherigen LabVIEW-Version und LabVIEW 2017 können gelegentlich zu Fehlern in VIs führen. Zum schnellen Erkennen und Korrigieren fehlerhafter VIs in LabVIEW 2017 führen Sie die folgenden Schritte aus:

1. Um durch das Upgrade verursachte VI-Fehler zu erkennen, vergleichen Sie das im vorherigen Schritt erzeugte Massenkompilierungsprotokoll mit dem Protokoll, das Sie beim Prüfen der aktuellen Funktionsweise der VIs erzeugt haben.
2. Um festzustellen, ob die VI-Fehler durch LabVIEW-Updates verursacht worden sind, lesen Sie den Abschnitt *Hinweise zur Kompatibilität beim Upgrade* in diesem Dokument.

## c. Erkennen und Korrigieren von Funktionsunterschieden

Auch wenn NI erhebliche Anstrengungen zur Wahrung der Rückwärtskompatibilität Ihrer VIs unternimmt, kommt es dennoch aufgrund von Verbesserungen oder Fehlerkorrekturen hin und wieder zu Funktionsunterschieden. Mit den folgenden Tools kann auf einfache Weise ermittelt werden, ob Ihre VIs von Änderungen an der neuen LabVIEW-Version betroffen sind:

- **Upgrade VI Analyzer Tests**—Mit Hilfe dieser Tests werden viele der durch die Aktualisierung verursachten Funktionsunterschiede in einer Vielzahl von VIs auf effiziente Weise erkannt. Führen Sie zum Herunterladen und Verwenden dieser Tests folgende Schritte aus:
  1. Laden Sie die Upgrade VI Analyzer Tests für alle Nachfolgeversionen Ihrer aktuellen LabVIEW-Version herunter. Zur Download-Seite für diese Tests gelangen Sie, indem Sie [ni.com/info](http://ni.com/info) besuchen und den Infocode `analyzevi` eingeben.
  2. Klicken Sie zum Starten der Tests auf **Werkzeuge»VI-Analyseprogramm»VIs analysieren** und starten Sie eine neue VI-Analyse-Aufgabe. Zur Analyse eines gesamten Projekts wählen Sie den entsprechenden Menüpunkt aus dem **Projekt-Explorer** aus.
  3. Zum Beheben der Probleme lesen Sie den Abschnitt *Hinweise zur Kompatibilität beim Upgrade* für die LabVIEW-Version, in der die Tests fehlgeschlagen sind. Wenn z. B. mit den LabVIEW 2014 Upgrade VI Analyzer Tests ein potentieller Funktionsunterschied erkannt wird, lesen Sie den Abschnitt *Upgrade von LabVIEW 2013*.
- **Upgrade-Dokumentation**
  - Abschnitt *Hinweise zur Kompatibilität beim Upgrade* dieses Dokuments—Listet Änderungen auf, die die Funktion Ihrer VIs beeinflussen können. Lesen Sie den Unterabschnitt für jede LabVIEW-Version beginnend bei Ihrer bisherigen Version.



**Tipp** Zum schnellen Auffinden von veralteten Objekten und anderen Objekten, die im Abschnitt *Hinweise zur Kompatibilität beim Upgrade* erwähnt werden, öffnen Sie Ihre aktualisierten VIs und wählen Sie **Bearbeiten»Suchen und Ersetzen** aus.

- Known Issues in LabVIEW 2017—Führt alle Programmfehler auf, die kurz vor der Veröffentlichung von LabVIEW 2017 erkannt wurden. Zu dieser Liste gelangen Sie, indem Sie [ni.com/info](http://ni.com/info) besuchen und den Infocode `lv2017ki` eingeben. Unter *Upgrade - Behavior Change* und *Upgrade - Migration* (falls vorhanden) können Sie sehen, welche Behelfslösungen für die einzelnen Probleme vorgeschlagen werden.
- Dokumentation für Module und Toolkits—Listet Upgrade-Probleme bei bestimmten Modulen und Toolkits auf, beispielsweise dem LabVIEW FPGA und LabVIEW Real-Time Module.

- Readme-Dateien zu Treibern—Enthalten Listen mit möglichen Upgrade-Problemen. Die Readme-Datei befindet sich auf dem Installationsdatenträger des Treibers.



**Tipp** Ob Funktionsunterschiede durch das Treiber- oder LabVIEW-Upgrade auf LabVIEW 2017 verursacht wurden, erfahren Sie durch Testen der VIs in Ihrer bisherigen LabVIEW-Version.

- **Ihre eigenen Tests**—Führen Sie in LabVIEW 2017 die gleichen Tests wie in der bisherigen LabVIEW-Version durch. Wenn Ihnen Funktionsunterschiede auffallen, ermitteln Sie anhand der Upgrade-Dokumentation die Ursache dafür.

## Fehlersuche beim Upgrade

In der *LabVIEW-Hilfe* finden Sie unter **Upgrade auf LabVIEW 2017»Fehlersuche beim Upgrade** Informationen zu den folgenden Themen:

- Suchen nach fehlenden Modul- oder Toolkit-Funktionen
- Suchen nach fehlenden SubVIs, Paletten und Eigenschaftsknoten
- Feststellen, warum sich VIs aus älteren Versionen nicht in LabVIEW 2017 öffnen lassen
- Ermitteln der Versionen installierter Softwareprodukte von National Instruments
- Zurücksetzen von VIs auf eine Vorgängerversion von LabVIEW

## Hinweise zur Kompatibilität beim Upgrade

---

In den folgenden Abschnitten werden versionsspezifische Änderungen an LabVIEW beschrieben, die in bestehenden VIs zu Fehlern oder Funktionsunterschieden führen können.

Bekannte Funktionsprobleme, Kompatibilitätsprobleme sowie kurz vor Veröffentlichung hinzugefügte Funktionen von LabVIEW 2017, die nicht anderweitig dokumentiert sind, werden in der Datei `readme.html` im `labview`-Verzeichnis beschrieben.

## Upgrade von LabVIEW 2012 oder älteren Versionen

Informationen zu möglichen Kompatibilitätsproblemen beim Upgrade von LabVIEW 2012 oder älteren Versionen auf LabVIEW 2017 finden Sie auf der Website von NI unter `ni.com/info` nach Eingabe des Infocodes `upnote13de`. Weitere mögliche Upgrade-Probleme sind im vorliegenden Dokument unter *Upgrade von LabVIEW x* beschrieben.

## Upgrade von LabVIEW 2013

Beim Upgrade von LabVIEW 2013 auf LabVIEW 2017 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten. Alle weiteren potentiellen Probleme beim Upgrade von LabVIEW sind in den Abschnitten *Upgrade von LabVIEW 2014*, *Upgrade von LabVIEW 2015* und *Upgrade von LabVIEW 2016* dieses Dokuments beschrieben.

### Geänderte Funktionsweise der Funktion "String nach Pfad"

In LabVIEW 2014 und neueren Versionen gibt die Funktion "String nach Pfad" ungeachtet der Groß- oder Kleinschreibung von `<Kein Pfad>` immer die Konstante `<Kein Pfad>` aus. Sie können also `<kein Pfad>` oder `<Kein Pfad>` am **String**-Eingang angeben. Die Arbeitsweise der Funktion "String nach Pfad" in älteren LabVIEW-Versionen ist in der folgenden Tabelle beschrieben.

LabVIEW 2012 und 2013	LabVIEW 2011 und älter
Die Funktion "String nach Pfad" gibt keine Konstante des Typs <Kein Pfad> aus. Wenn Sie eine Variation des Strings <Kein Pfad> angeben, gibt die Funktion statt der Konstante <Kein Pfad> einen Pfad zu einem Verzeichnis namens <Kein Pfad> aus.	Die Funktion "String nach Pfad" ignoriert wie in LabVIEW 2014 und neueren Versionen die Groß- und Kleinschreibung des Strings <Kein Pfad> und gibt immer die Konstante <Kein Pfad> aus. Unabhängig davon, ob Sie <kein Pfad> oder <Kein Pfad> angeben, gibt die Funktion immer die Konstante <Kein Pfad> aus.

## Vergleichen und Aktualisieren von Typdefinitionen

Der Kontextmenüpunkt **Mit Typdefinition vergleichen und aktualisieren** ist in LabVIEW 2014 an die Stelle des bisherigen Menüpunkts **Aktualisierung von Typdefinition** getreten.

## Veraltete VIs, Funktionen und Knoten

Folgende VIs, Funktionen und Knoten werden von LabVIEW 2014 und neueren LabVIEW-Versionen nicht mehr unterstützt:

### AppleEvent-VIs

(OS X) LabVIEW-Versionen ab 2014 unterstützen keine AppleEvent-VIs mehr. Zum Datenaustausch mit LabVIEW-externen OS-X-Anwendungen wird jetzt das VI "Run AppleScript Code" auf der Palette "Libraries & Executables" benötigt. Beim Laden eines VIs mit einem der folgenden AppleEvent-VIs können Fehler auftreten, und das VI ist evtl. nicht ausführbar:

- "AESend Do Script"
- "AESend Finder Open"
- "AESend Open"
- "AESend Open Document"
- "AESend Print Document"
- "AESend Quit Application"
- "Get Target ID"
- "AESend Abort"
- "AESend Close"
- "AESend Open, Run, Close"
- "AESend Run"
- "AESend VI Active?"
- "AECreat Comp Descriptor"
- "AECreat Descriptor List"
- "AECreat Logical Descriptor"
- "AECreat Object Specifier"
- "AECreat Range Descriptor"
- "AECreat Record"
- "AESend"
- "Make Alias"

### Akteur-Framework-VIs

LabVIEW 2014 und neuere LabVIEW-Versionen unterstützen das VI "Akteur:Akteur starten" nicht. Verwenden Sie stattdessen das VI "Akteur:Hauptakteur starten" oder das VI Akteur:Unterakteur starten.

### VIs "Eingangs-Port" und "Ausgangs-Port"

Die VIs "Eingangs-Port" und "Ausgangs-Port" werden in LabVIEW 2014 und neueren LabVIEW-Versionen nicht unterstützt.

## Veraltete Eigenschaften, Methoden und Ereignisse

LabVIEW 2014 und neuere LabVIEW-Versionen unterstützen die Methode "VI ermitteln:Hilfe-Info (alt)" der Klasse "Anwendung" nicht. Verwenden Sie stattdessen die Methode "VI ermitteln:Hilfe-Info", um von der Seite **Dokumentation** des Dialogfelds **Eigenschaften für VI** Angaben zu einem VI abzufragen.

## Upgrade von LabVIEW 2014

Beim Upgrade von LabVIEW 2014 auf LabVIEW 2017 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten. Alle weiteren potentiellen Probleme beim Upgrade von LabVIEW sind in den Abschnitten *Upgrade von LabVIEW 2015* und *Upgrade von LabVIEW 2016* dieses Dokuments beschrieben.

### Erkennen von Pufferzuteilungen in LabVIEW-Anwendungen

Zu den Neuerungen in LabVIEW 2014 Service Pack 1 und neueren Versionen gehört das Fenster **Pufferzuteilung aufschlüsseln** mit Informationen zur Puffervergabe innerhalb einer LabVIEW-Anwendung. Klicken Sie zur Anzeige dieses Dialogfelds auf **Werkzeuge»Profil»Pufferzuteilung aufschlüsseln**.

### Hyperlinks in freien Beschriftungen

LabVIEW 2015 und neuere Versionen erkennen URLs in freien Beschriftungen und kennzeichnen diese automatisch durch blauen unterstrichenen Text. Beim Upgrade von LabVIEW 2014 oder älteren Versionen werden URLs in freien Beschriftungen jedoch nicht automatisch in Hyperlinks umgewandelt. Zum Aktivieren von Hyperlinks in Frontpanel-Beschriftungen klicken Sie mit der rechten Maustaste auf eine freie Beschriftung und setzen Sie ein Häkchen vor dem Menüpunkt **Hyperlinks aktivieren**. Hyperlinks in Blockdiagrammbeschriftungen können nicht deaktiviert werden.

### Veraltete VIs, Funktionen und Knoten

Folgende VIs werden von LabVIEW 2015 und neueren LabVIEW-Versionen nicht mehr unterstützt:

- **Aus Tabellenkalkulationsdatei lesen**—Verwenden Sie stattdessen das VI "Tabelle mit Trennzeichen lesen".
- **In Tabellenkalkulationsdatei schreiben**—Verwenden Sie stattdessen das VI "Tabelle mit Trennzeichen schreiben".

## Upgrade von LabVIEW 2015

Beim Upgrade von LabVIEW 2015 auf LabVIEW 2017 kann das nachfolgend aufgeführte Kompatibilitätsproblem auftreten. Alle weiteren potentiellen Probleme beim Upgrade von LabVIEW sind im Abschnitt *Upgrade von LabVIEW 2016* dieses Dokuments beschrieben.

In LabVIEW 2016 und neueren LabVIEW-Versionen enthält das Dialogfeld **Schnelleinfüge-Einstellungen** eine Liste mit Standard-Bedientasten für Frontpanel- und Blockdiagrammobjekte. Wenn Sie von LabVIEW 2015 oder älteren Versionen auf LabVIEW 2016 umsteigen, werden in der älteren Version hinzugefügte Schnell Tasten nicht automatisch in die Liste der neuen Version übernommen.

## Upgrade von LabVIEW 2016

Beim Upgrade von LabVIEW 2016 auf LabVIEW 2017 kann das nachfolgend aufgeführte Kompatibilitätsproblem auftreten.

### Änderungen in der Funktionsweise bei Akteur-Framework-VIs


In LabVIEW-Versionen bis 2016 gibt der Unterakteur für einen fehlgeschlagenen Start aufgrund eines Fehlers in der Methode "Init vor Start" eine Fehlermeldung aus und sendet die Nachricht "Letzte



Bestätigung" an den aufrufenden Akteur. In LabVIEW 2017 gibt der Unterakteur einen Fehler aus, ohne die Meldung "Letzte Bestätigung" an den aufrufenden Akteur zu senden.

## Änderungen und Neuerungen in LabVIEW 2017

---

Mit dem Symbol  gekennzeichnete Neuerungen stammen von Vorschlägen aus dem LabVIEW Idea Exchange. Zum NI Idea Exchange gelangen Sie über [ni.com/info](http://ni.com/info) nach Eingabe des Infocodes `ex3gus`.

In der Datei `readme.html` im `labview`-Verzeichnis finden Sie Informationen zu bekannten Funktionsproblemen von LabVIEW 2017 sowie zu Fehlerkorrekturen, Kompatibilitätsproblemen und kurz vor Veröffentlichung hinzugefügten Funktionen.


### Reduzierte Lade- und Kompilierzeit für VIs

**(Windows)** Mit LabVIEW 2017 wurde für sowohl die LabVIEW Entwicklungsumgebung als auch die LabVIEW-Runtime-Engine auf einen leistungsstärkeren Compiler aktualisiert. Durch dieses Upgrade wird die Gesamtladezeit für VIs und die Kompilierzeit reduziert.

### Beibehalten von Verbindungen beim Verschieben von Objekten

Beim Verschieben von Objekten aus und in Blockdiagramm-Strukturen werden die Verbindungen in LabVIEW 2017 automatisch aufrechterhalten. Wenn ein Objekt, das aus einer Struktur herausgezogen wird, mit einem Objekt innerhalb der Struktur verbunden ist, werden von LabVIEW Tunnel erstellt (bzw. entfernt), um die Verbindung beizubehalten. Mit der Taste <W> können Sie das automatische Verbinden ein- oder ausschalten.

### Datentypflexible VIs

 LabVIEW 2017 enthält datentypflexible VIs (`.vim`), die inline in aufrufende VIs eingefügt werden. In einem datentypflexiblen VI kann jeder Anschluss an den entsprechenden Eingangsdatentyp angepasst werden. Mit datentypflexiblen VIs kann die gleiche Operation an verschiedenen zulässigen Datentypen durchgeführt werden. Das Speichern einer Kopie des VIs für jeden Datentyp wird dadurch überflüssig.

Ein datentypflexibles VI ähnelt einem polymorphen VI, ist jedoch beim Bestimmen von zulässigen Datentypen anpassungsfähiger. Ein polymorphes VI verwendet eine vordefinierte Liste von zulässigen Datentypen. Ein datentypflexibles VI ermittelt anhand seiner Implementierung, ob es den angelegten Datentyp unterstützt.

Datentypflexible VIs haben die Dateierweiterung `.vim`. Sie können ein datentypflexibles VI erstellen, indem Sie **Datei»Neu** öffnen und aus dem Dialogfeld **Neu** die Option **Datentypflexibles VI** auswählen. Wenn Sie ein vorhandenes VI in ein datentypflexibles VI umwandeln möchten, speichern Sie die Datei mit der Dateierweiterung `.vim`.



**Hinweis** Sie können nur Standard-VIs in datentypflexible VIs umwandeln. Polymorphe VIs, globale VIs und XControl-Leistungsmerkmale können nicht in datentypflexible VIs umgewandelt werden.

### In LabVIEW enthaltene datentypflexible VIs

Die folgenden datentypflexiblen VIs sind zur Verwendung in Ihren LabVIEW-Anwendungen verfügbar. Die Symbole für in LabVIEW vorhandene datentypflexible VIs haben einen orangefarbenen Hintergrund.

- Palette "Array"
  - **"Array-Element dekrementieren"**—Subtrahiert 1 vom angegebenen Element eines 1D-Arrays. Wenn es sich um ein Zeitstempel-Array handelt, dekrementiert das VI das Element um eine Sekunde.

- **"Array-Element inkrementieren"**—Addiert 1 zum angegebenen Element eines 1D-Arrays. Wenn es sich um ein Zeitstempel-Array handelt, inkrementiert das VI das Element um eine Sekunde.
- **"1D-Array mischen"**—Ordnet die Elemente eines 1D-Arrays in pseudozufälliger Reihenfolge neu an.
- **"2D-Array mischen"**—Ordnet die Elemente eines 2D-Arrays in pseudozufälliger Reihenfolge neu an.
- **"2D-Array sortieren"**—Ordnet die Zeilen und Spalten eines 2D-Arrays erneut an, indem die Elemente in der angegebenen Spalte oder Zeile in aufsteigender Reihenfolge sortiert werden.
- Palette "Vergleich"
  - **"Wert geändert"**—Gibt TRUE aus, wenn das VI zum ersten Mal aufgerufen wird oder wenn sich der Eingangswert vom Wert beim letzten Aufruf unterscheidet.
- Palette "Konvertierung"
  - **"Zahl nach Enum"**—Sucht nach einem Enum-Wert, der mit dem angegebenen Wert übereinstimmt, und gibt den entsprechenden Enum-Eintrag aus.
- Palette "Timing"
  - **"Datenfluss aufhalten"**—Verzögert den Datenfluss der Verbindung eine angegebene Zeit lang.

Weitere Informationen zu diesen Programmfunktionen erhalten Sie unter **Grundlagen»Erstellen von VIs und SubVIs»Allgemeines»Erstellen von modularem Programmcode»Datentypflexible VIs** in der *LabVIEW-Hilfe*.

Ein Beispiel zur Verwendung von datentypflexiblen VIs finden Sie im Projekt `labview\examples\Malleable VIs\Basics\Malleable VIs Basics.lvproj`.

*[Idee wurde vom NI-Diskussionsforen-Mitglied DanyAllard eingereicht]*

## Neue und geänderte VIs und Funktionen

In LabVIEW 2017 sind folgende Neuerungen an VIs und Funktionen vorgenommen worden:

### Schreibgeschützter Zugriff auf Datenwertreferenzen

Der Rahmenknoten Datenwertreferenz lesen / schreiben der Inplace-Elementstruktur ermöglicht jetzt den schreibgeschützten Zugriff auf eine Datenwertreferenz. Klicken Sie dazu mit der rechten Maustaste auf den rechten Rahmenknoten der Struktur und wählen Sie **Parallele schreibgeschützte Zugriffe zulassen** aus. Wenn der Rahmenknoten rechts mit keinem Objekt verbunden ist, lässt LabVIEW mehrere gleichzeitige schreibgeschützte Operationen zu, ohne die Datenwertreferenz zu modifizieren.

### Neue Kanalvorlagen

In LabVIEW 2017 ist die Ereignis-Messenger-Kanalvorlage neu hinzugekommen. Mit ihrer Hilfe lassen sich Daten von mehreren schreibenden Objekten an eine oder mehrere Ereignisstrukturen übertragen. Bei jedem Schreibvorgang in den Kanal wird ein Ereignis ausgelöst. Mit Hilfe des Ereignis-Messenger-Kanals kann die Kanalsyntax mit der Syntax zur Steuerung Ihrer Ereignisse der Benutzeroberfläche und erzeugter Ereignisse kombiniert werden. Ein Anwendungsbeispiel für den Ereignis-Messenger-Kanal finden Sie im Projekt `labview\examples\Channels\Event Messenger\Channel - Event Messenger.lvproj`.

## Neue und geänderte Klassen, Eigenschaften, Methoden und Ereignisse

LabVIEW 2017 umfasst Änderungen an der Methode "VI-Abhängigkeiten ermitteln (Namen und Pfade)". Der Parameter **Express-VIs beibehalten?** wurde in **Express-VIs und datentypflexible VIs beibehalten?**

umbenannt. Wenn **Express-VIs und datentypflexible VIs beibehalten?** FALSE ist (Standardeinstellung), gibt LabVIEW die Namen der versteckten Instanz-VIs aus, die den Express-VIs und datentypflexiblen VIs zugrunde liegen. Bei TRUE gibt LabVIEW die Express-VIs und datentypflexiblen VIs als Abhängigkeiten aus. Um während der Bearbeitung existierende Abhängigkeiten abzufragen, setzen Sie **Express-VIs und datentypflexible VIs beibehalten?** auf TRUE. Um während der Ausführung existierende Abhängigkeiten zu ermitteln, setzen Sie **Express-VIs und datentypflexible VIs beibehalten?** auf FALSE. Unabhängig von dieser Einstellung schließt LabVIEW die SubVIs der Instanz-VIs als Abhängigkeiten des VIs ein, auf das verwiesen wurde.

## Verbesserungen am Application Builder

In LabVIEW 2017 wurden folgende Verbesserungen am LabVIEW Application Builder und an den Build-Spezifikationen vorgenommen:

### Rückwärtskompatibilität der LabVIEW-Runtime-Engine

In bisherigen LabVIEW-Versionen konnten Binärdateien und VIs aus älteren LabVIEW-Versionen nur mit erneutem Kompilieren geladen und ausgeführt werden. Beginnend mit LabVIEW 2017 entfällt diese Notwendigkeit. Alle in künftigen LabVIEW-Versionen erzeugten Binärdateien und VIs können dann beispielsweise in LabVIEW 2017 ohne erneute Kompilierung geladen werden. EXE-Dateien, DLLs und komprimierte Projektbibliotheken profitieren von dieser Neuerung.

Um die Rückwärtskompatibilität für Binärdateien zu aktivieren, markieren Sie das folgende Auswahlfeld auf der Seite **Fortgeschritten** im jeweiligen Dialogfeld (abhängig von Ihren Build-Spezifikationen):

Build-Spezifikation	Dialogfeld	Auswahlfeld
Eigenständige Anwendung (EXE)	Eigenschaften für Anwendung	Künftigen LabVIEW-Runtime-Versionen das Ausführen dieser Anwendung gestatten
Komprimierte Projektbibliotheken	Eigenschaften für komprimierte Bibliothek	Künftigen LabVIEW-Versionen das Laden dieser komprimierten Bibliothek gestatten
DLL	Eigenschaften für DLL	Künftigen LabVIEW-Runtime-Versionen das Laden dieser DLL gestatten

Bei Build-Spezifikationen, die Sie in LabVIEW 2017 oder einer neueren Version erzeugen, sind diese Optionen standardmäßig aktiviert. Sie können diese Optionen deaktivieren, wenn Sie eine Build-Spezifikation an eine bestimmte Version von LabVIEW binden möchten. Das Deaktivieren dieser Optionen verhindert Änderungen an den Leistungsprofilen, wodurch aus Compiler-Upgrades resultierende unerwartete Probleme vermieden werden. Bei Real-Time-Anwendungen werden diese Optionen zwar nicht in den Dialogfeldern angezeigt, die Funktion ist jedoch standardmäßig aktiviert.

### Verbesserungen an Aufrufen zwischen LabVIEW und anderen Programmiersprachen

In LabVIEW 2017 wurden signifikante Verbesserungen an der Arbeitsweise und Stabilität von in LabVIEW erzeugten DLLs vorgenommen. Dies betrifft insbesondere das Aufrufen der DLLs von LabVIEW und anderen Programmiersprachen aus. Beispielsweise können in LabVIEW erzeugte DLLs jetzt von C-Anwendungen in multithreadfähigen Systemen aufgerufen werden. Diese Verbesserungen verhindern darüber hinaus, das beim Aufrufen der DLLs von LabVIEW aus Blockaden und Atomaritätsverstöße auftreten.

Aktivieren Sie zum Verwenden dieses Funktionsumfangs im Dialogfeld **Eigenschaften für DLL** auf der Seite **Fortgeschritten** die Option **VIs in privatem Ausführungssystem ausführen**. Für neue Build-Spezifikationen ist diese Option standardmäßig aktiviert. Für Build-Spezifikationen, die aus LabVIEW 2016 oder älteren LabVIEW-Versionen stammen, ist diese Option per Standardeinstellung deaktiviert, um die beabsichtigte Funktionsweise der DLLs zu wahren. Auf diese Weise wird zum

Beispiel verhindert, dass für die Einzel-Thread-Ausführung vorgesehene DLLs, die in LabVIEW erzeugt wurden, von einem Programm in einer Multithreading-Umgebung aufgerufen werden. (**NI Linux Real-Time**) Diese Option ist standardmäßig für Linux-RT-Zielsysteme deaktiviert, um potentielle Fluktuationen in der Ausführungsgeschwindigkeit zu vermeiden.

## Änderungen und Neuerungen in vorherigen LabVIEW-Versionen

---

Alle neuen Funktionen, die seit Ihrer bestehenden LabVIEW-Version hinzugekommen sind, finden Sie in den Upgrade-Hinweisen der jeweiligen Versionen. Zum Öffnen dieser Dokumente besuchen Sie die Seite [ni.com/info](http://ni.com/info) und geben Sie einen der folgenden Infocodes ein:

- *Hinweise zum Upgrade auf LabVIEW 2013*—[upnote13de](#)
- *Hinweise zum Upgrade auf LabVIEW 2014*—[upnote14de](#)
- *Hinweise zum Upgrade auf LabVIEW 2015*—[upnote15de](#)
- *Hinweise zum Upgrade auf LabVIEW 2016*—[upnote16de](#)

Weitere Informationen zu Marken von NI finden Sie in den *NI Trademarks and Logo Guidelines* auf [ni.com/trademarks](http://ni.com/trademarks). Sonstige hierin erwähnte Produkt- und Firmenbezeichnungen sind Marken oder Handelsnamen der jeweiligen Unternehmen. Nähere Informationen über den Patentschutz von Produkten/Technologien von NI finden Sie unter **Hilfe»Patente** in Ihrer Software, in der Datei `patents.txt` auf Ihrem Datenträger oder unter *National Instruments Patent Notice* auf der Website [ni.com/patents](http://ni.com/patents). Informationen zu Endbenutzer-Lizenzverträgen (EULAs) und Rechtshinweisen von Drittanbietern finden Sie in der Readme zu Ihrem Produkt von National Instruments. Informationen zu den Global-Trade-Compliance-Richtlinien von NI sowie zu Bezugsquellen für relevante HTS-Codes, ECCNs und andere Import-/Exportangaben finden Sie auf [ni.com/legal/export-compliance](http://ni.com/legal/export-compliance) unter der Überschrift *Export Compliance Information*. NI übernimmt weder ausdrücklich noch stillschweigend irgendeine Gewährleistung für die Richtigkeit oder Vollständigkeit der in diesem Dokument enthaltenen Informationen und haftet nicht für Fehler. Kunden aus US-Regierungsbehörden: Die in diesem Dokument enthaltenen Informationen wurden auf private Kosten entwickelt und unterliegen den anwendbaren beschränkten Rechten und beschränkten Datenrechten, die in FAR 52.227-14, DFAR 252.227-7014 und DFAR 252.227-7015 dargelegt sind.